Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems

David Formby*, Preethi Srinivasan*, Andrew Leonard†, Jonathan Rogers†, Raheem Beyah*
*School of Electrical and Computer Engineering
Georgia Institute of Technology
djformby@gatech.edu, preethisrinivasan@gatech.edu, rbeyah@ece.gatech.edu
†School of Mechanical Engineering
Georgia Institute of Technology
aleonard31@gatech.edu, jonathan.rogers@me.gatech.edu

***Abstract*—Industrial control system (ICS) networks used in critical infrastructures such as the power grid present a unique set of security challenges. The distributed networks are difficult to physically secure, legacy equipment can make cryptography and regular patches virtually impossible, and compromises can result in catastrophic physical damage. To address these concerns, this research proposes two device type fingerprinting methods designed to augment existing intrusion detection methods in the ICS environment. The first method measures data response processing times and takes advantage of the static and low-latency nature of dedicated ICS networks to develop accurate fingerprints, while the second method uses the physical operation times to develop a unique signature for each device type. Additionally, the physical fingerprinting method is extended to develop a completely new class of fingerprint generation that requires neither prior access to the network nor an example target device. Fingerprint classification accuracy is evaluated using a combination of a real world five month dataset from a live power substation and controlled lab experiments. Finally, simple forgery attempts are launched against the methods to investigate their strength under attack.**

I. Introduction

Fingerprinting devices on a target network, whether it is based on their software or hardware, can provide network administrators with mechanisms for intrusion detection or enable adversaries to conduct surveillance in preparation for a more sophisticated attack. In the context of industrial control systems (ICS), where a cyber-based compromise can lead to physical harm to both man and machine, these mechanisms become even more important. An attacker intruding on an ICS network can theoretically inject false data or commands and drive the system into an unsafe state. Example consequences of such an intrusion can range from widespread blackouts in the power grid [24] to environmental disasters caused by tampering with systems carrying water, sewage [3], oil, or natural gas. These false data and command injections could be thwarted using strong cryptographic protocols that provide integrity and authentication guarantees, but in ICS networks it is often infeasible to upgrade legacy equipment to provide them due to lack of processing power, devices being in remote locations, and the critical nature of the systems that must be online at all times. In fact, some vendors do not even support the functionality of upgrading devices to install critical patches. When our previous research found vulnerabilities in several power system devices and they were reported to ICS-CERT, the resulting official advisory for one of the products stated that "There is no method to update [Product A] devices released prior to October 2014 [1]." Since adding cryptography to resource limited devices and keeping them patched is infeasible and sometimes just impossible, alternative methods such as fingerprinting must be used to provide security and intrusion detection.

While device fingerprinting is a well-studied topic with several solutions already proposed, none of them are properly suited for the ICS environment. Active fingerprinting techniques can achieve high accuracy detection of operating systems and server versions, but require probing the network with specially crafted packets. This solution is undesirable in an ICS environment where devices are performing time-critical functions and administrators would rather not risk even the small chance of a port scan crashing the legacy devices and resulting in critical system downtime that includes loss of revenue and potentially life-threatening situations for affected customers such as hospitals. Therefore passive techniques are more suited, but they usually provide limited useful information or require special equipment or TCP options enabled.

This paper presents some of the defining characteristics of ICS networks and discusses how to use them to develop two new fingerprinting approaches that perform uniquely well in the ICS environment, where the two primary functions are *data acquisition* and *control* and are carried out over supervisory control and data acquisition (SCADA) protocols such as DNP3, Modbus, and IEC 61850 GOOSE. Our first approach takes advantage of the data acquisition functions by using the interaction between the application layer responses and transport layer acknowledgments to obtain measures estimating the speed and workload of a particular intelligent electronic device (IED). Due to the unique properties of ICS networks, the distributions of these measurements are constant within device types and software configurations allowing network

NDSS '16, 21-24 February 2016, San Diego, CA, USA
Copyright 2016 Internet Society, ISBN 1-891562-41-X
http://dx.doi.org/10.14722/ndss.2016.23142

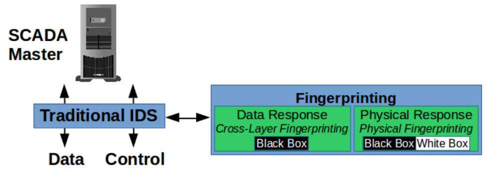


Fig. 1. The two novel fingerprinting methods can work together to augment traditional intrusion detection

administrators to passively detect changes in the configuration or spoofed communication. Throughout this work we refer to this technique as cross-layer fingerprinting. Our second fingerprinting approach uses the control aspects of ICS environments to generate signatures from the physical operations being taken by the physical devices on the network. Even though two relays or valves from different vendors may have similar ratings, there will always be physical variations in their construction resulting in fundamental differences in their operation times. These differences are then used to identify device types or spoofed command responses, which we call physical fingerprinting. When used together as illustrated in Figure 1, these two methods can achieve device fingerprinting from software, hardware, and physics based perspectives and provide a strong supplement to more traditional intrusion detection systems (IDS) in the ICS environment.

The fingerprint (or signature) of a device can be represented as a probability density function (PDF) of the response times described above. To generate these PDFs, one of three approaches can be used: white box, black box, and gray box modeling. In a white box approach, a dynamic model of the device is constructed from first principles and model parameters identified from CAD drawings, source code, physical measurements, etc. without ever seeing any true samples from the system. The simulated behavior is then used to create a PDF by varying model parameters using an uncertainty distribution. In a black box approach, the PDF is constructed strictly from experimental data without any dynamic modeling, requiring a significant amount of experimental measurements but little knowledge of the underlying system. *Until now, this approach has been the only method used by all previous fingerprinting work*. Finally, in a gray box approach, a dynamic model is first constructed and the resulting PDF is then refined based on experimental measurements. White box modeling is best suited for when a system's internal details are accessible but access to experimental measurements is restricted. Black box modeling performs best when experimental measurements are easily available and especially when the system is proprietary or too complex to model. Finally, gray box approaches are most advantageous when the basic characteristics of a software or hardware design are known, but there is some uncertainty in model structure or parameters that can only be dealt with through experimental observations [17].

Due to the abundance of measurements in the available dataset and lack of proprietary source code, the data acquisition fingerprinting method proposed here, called cross-layer fingerprinting, focuses on a black box modeling approach. In the case of the physical fingerprinting technique, there are some devices where the operations occur so rarely that collecting enough real samples to generate an accurate fingerprint through black box modeling is completely infeasible. Additionally, there is such a wide variety of physical devices available and their costs are so prohibitive that creating a black box signature database offline is also infeasible. Therefore an alternative approach for signature generation must be used. This paper proposes a new class of fingerprint generation for physical fingerprinting based on white box modeling to allow an administrator to generate a usable device fingerprint without ever having access to the target device type or network. The white box generated physical fingerprint is then validated against the black box approach using an example control device.

The major contributions of this research include:

- Two novel fingerprinting approaches that take advantage of the unique characteristics of ICS devices
- A new class of fingerprint generation specific to ICS networks using "white box" modeling
- Performance analysis using both real world data from a power substation and controlled lab tests
- Evaluation of the methods under simple forgery attacks for different classes of adversary

The remainder of this paper is organized as follows. Related work in the area of device fingerprinting and intrusion detection in ICS is presented in Section II. The assumptions and threat model addressed by this work are presented in Section III and the details of the cross-layer and physical fingerprinting methods using black box modeling are discussed in Section IV. Section V provides an explanation of the extension of the physical fingerprinting technique via the use of synthetic signatures generated from white box modeling. Finally, the performance and limitations of the techniques are discussed in Section VI, and the results and future work are summarized in Section VII.

II. Related Work

Device fingerprinting methods are usually classified into active or passive techniques depending on whether they actively probe a device with specially crafted packets or passively monitor network traffic to develop the fingerprint.

One of the oldest and most well known fingerprinting tools, Nmap, uses active fingerprinting techniques to gather information about devices on a network [2]. By sending a series of specific requests, Nmap determines the OS and server versions running on a machine based on how the device responds. While this tool is invaluable for both pen-testers and attackers on a "normal" network, it has limited use in an ICS network where active methods are not as desirable.

For passive fingerprinting, a variety of techniques exist that provide both device type fingerprinting and individual device fingerprinting. One example is the open source p0f tool, which passively examines TCP and HTTP header fields to determine information about a client, such as OS and browser version [26]. The first attempt at formalizing methods for active and passive fingerprinting of network protocols was published in 2006, when authors used parametrized extended finite state machine (PEFSMs) to model the behavior of different protocol implementations [19]. Determining software versions is of some use, but identifying individual devices on a network based on their hardware is even more useful, which for example could be used for tracking a device across the Internet or intrusion detection. Using both passive and active techniques, Kohno et al. produced the first such work on individual device fingerprinting in 2005 by examining TCP timestamps to detect individual device clock skew [13].

Other passive fingerprinting research has focused on various timing aspects of network traffic to fingerprint devices and device types. In 2010 researchers were able to use wavelet analysis on passively observed traffic flowing through access points to accurately identify each access point [12]. The next year, another paper was published that described a method for device fingerprinting based on models of the timing of a device's implementation of application layer protocols using Temporal Random Parametrized Tree Extended Finite State Machines (TR-FSMs) [11]. A third paper that used passive observations of network traffic timing to achieve device fingerprinting was published in 2014, and used distributions of packet inter-arrival times (IAT) to identify devices and device types [18].

Although these three papers all took different approaches to using passively observed network traffic timing to perform fingerprinting, they are all infeasible for implementation in an ICS network. The wavelet analysis approach was designed and tested only on wireless access points under heavy loads, a scenario that does not occur in ICS where wired communication is preferred for its reliability and data rates are relatively low. The method using TR-FSMs only looks at application layer behaviors and requires a large database of all possible sessions. Finally, the method using distributions of IATs requires a large number (at least 2500) of training samples to achieve accurate results, but with some devices on ICS networks being polled at an interval as large as a few seconds, this would result in unacceptably slow operation. Another technique was developed that used timing measurements of USB enumerations to fingerprint host devices [4], but this is also impractical in the ICS environment where most devices do not have USB interfaces and where it is desirable to passively fingerprint all devices on the network at once rather than driving out to remote locations to fingerprint each individual device.

Another unique approach to passive device fingerprinting relevant to this paper focused on the physical layer of device communication, rather than the higher layers. Specifically, researchers were able to use amplitude and phase measurements of the signals generated by Wi-Fi radios to identify individual devices [21]. This may have been the first work to use physical measurements to fingerprint devices, but it still is not feasible in ICS networks where Wi-Fi devices are rarely used.

The two methods presented in this paper overcome the limitations of the previous work on device fingerprinting by providing higher accuracy results using techniques specially suited for ICS networks. The first method improves on the more traditional timing based approaches by using network traffic measurements that are unique to ICS devices, and the second proposed method extends the idea of physical layer fingerprinting to identifying ICS control devices based on the reported timings of their physical operations. Additionally, all previous fingerprinting work used black box methods that require access to an example target device. This research also overcomes this limitation by proposing a white box fingerprint generation approach that requires no previous access to example devices.

One of the primary uses of the two proposed fingerprinting techniques would be to augment existing IDS solutions, of which there is already a significant amount of previous work. The first attempt at tailoring IDS methods for ICS and SCADA systems was proposed by Idaho National Laboratories in 2008, and focused on monitoring traffic flows for regular patterns and understanding packets at the application layer to look for intrusions [23]. Some researchers have also approached the problem by modifying the popular Bro IDS software to perform specification based intrusion detection for common ICS protocols [16]. Others have attempted to model the states a process control system can enter and detect when a command might cause it to enter a critical state [10] [5]. These solutions are able to detect some types of attacks, but are unable to detect a class of stealthier ones called false data injection attacks. To address this, some methods have been proposed for power system state estimation [14] and for process control systems [6]. However, they are only useful in the context of power state estimation or where the process behind the control system can be accurately modeled. The fingerprinting methods proposed in this paper offer two novel approaches that are generic enough to be applied to most ICS networks and enable accurate detection of falsified data and control messages.

III. Threat Model, Assumptions, and Goals

Without loss of generality, this paper addresses the proposed methods in terms of the power grid with extensions to other ICS applications being easily made.

One of the unique challenges for ICS network security is the vast attack surface available due to the distributed nature of the networks. For example, the electric utility from which experimental data was gathered for this research covers an area of *2800 square miles with 35 substations*, where each substation serves as a point of entry to the network. With such a large area to cover, physical security is extremely difficult to achieve [20]. Therefore, we consider two different attacker models: 1) an outsider who is unable to gain physical access but has compromised a low powered node in the network with malware, and 2) an outsider who is feasibly able to gain physical access to the target network and use her own portable machine with standard laptop computing power. The first attacker model was chosen due to how vulnerable these devices are (as evidenced by the 30 year old TCP vulnerabilities found widespread in the power grid [9]) and because it was the method used on the most well known ICS attack to date, Stuxnet [15]. The second attacker model is realistic in the scenario of a widely distributed control system where physical security is difficult to achieve.

Figure 2 illustrates the different points of attack that an adversary can take advantage of when attacking a power substation network. He can either attack the communication infrastructure or one of the individual devices such as the remote terminal unit (RTU) or a programmable logic controller (PLC). Depending on where he attacks the network, the adversary can attempt to inject false data responses, false command responses, or both. As already discussed in the related work in Section II, false data and command injections such as

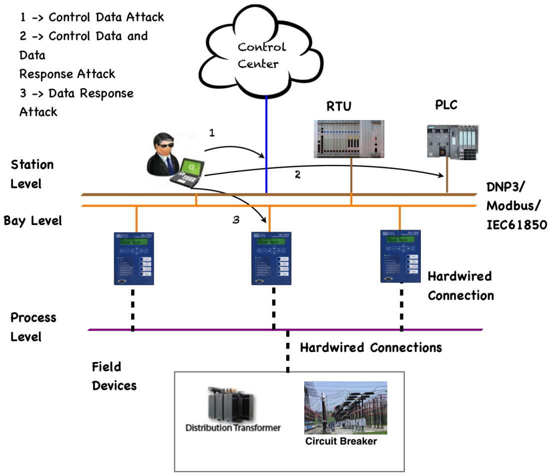


Fig. 2. Points of attack in a power substation network

these can have disastrous effects on the power grid. With this in mind, the goal of this research is to develop accurate fingerprinting methods to identify what *type* of device these responses are originating from as opposed to unique devices. Such methods could be crucial to distinguishing between responses originating from a legitimate IED, an adversary with a laptop who has gained access to the network, or a comprised IED posing as a different device on the network.

For a formal definition, assume the global set of all ICS devices G consists of products $D_{j,k}$, where j identifies the vendor and k signifies the model for each vendor's product. Given a sequence of observations O_i every device i on the network, the goal of the fingerprinting methods will be to identify which subset of G, specifically which $D_{j,k}$, those observations belong to.

IV. Overview of Device Fingerprinting Methods

Two of the properties that differentiate ICS networks from more traditional networks are their primary functions of data acquisition through regular polling for measurements and control commands. These properties hold true for all of the most critical ICS networks regardless of the underlying physical process, including the distribution of power, water, oil, and natural gas. The two methods proposed below take advantage of these unique properties and are explained using the power grid as a specific example. The first method is evaluated using data from a live power substation and verified with controlled lab experiments. The second method is evaluated only with lab experiments due to the relatively rare occurrence of operations in the given dataset[1], but it should be noted that other power grid networks and industries, such as oil and gas, have more frequent operations.

A. Method 1: Cross-layer Response Times

The first proposed fingerprinting method addresses the data acquisition half of SCADA systems by leveraging the interaction between regular polling of measurement data at the application layer with acknowledgments at the TCP layer to get an estimate of the time a device takes to process the request, and then develops a fingerprint for each device based on the distribution of these times. The timing diagram of how this measurement, which we call the cross-layer response time (CLRT), would be taken in a typical SCADA network is illustrated in Figure 3. It should be noted that since the CLRT measurement is based on the time between two consecutive packets from the same source to the same destination, it is independent of the round trip time between the two nodes.

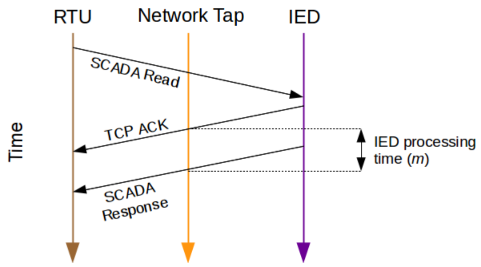


Fig. 3. Measurement of cross-layer response time

The fingerprint signature is defined by a vector of bin counts from a histogram of CLRTs where the final bin includes all values greater than a heuristic threshold. For a formal definition, let M be a set of CLRT measurements from a specific device, B define the number of bins in the histogram (and equivalently the number of features in the signature vector), and H signify the heuristic threshold chosen to be an estimate of the global maximum that CLRT measurements should ever take. We divide the range of possible values by thresholds t_i where $t_i = i\frac{H}{B-1}$, and define each element s_j of the signature vector by the following equation:

$$s_j = \begin{cases} |\{m : t_{j-1} \leq m < t_j, m \in M\}| & 0 < j < B \\ |\{m : m > H, m \in M\}| & j = B \end{cases} \quad (1)$$

1) Theory: The CLRT measurement is advantageous for fingerprinting ICS devices because it remains relatively static and its distribution is unique within device types and even software configurations. To understand why this is true for ICS devices, all of the factors which might affect this measurement must be considered.

Device Characteristics. ICS devices have simpler hardware and software architectures than general purpose computers because they are built to perform very specialized critical tasks and do little else. A typical modern-day computer now has fast multi-core processors in the range of 2-3GHz with significant caching, gigabytes of RAM, and context switching between the wide variety of processes running on the machine. In contrast, the ICS world is dominated by programmable logic controllers (PLCs) running on low powered CPUs in the tens to hundreds of MHz frequencies with little to no caching, tens to hundreds of megabytes of RAM and very few processes. With such limited computing power available, relatively small changes in programming result in observable timing differences. Depending on the desired task, different ICS device types are built with different hardware specifications (CPU frequencies, memory and bus speeds) [18] as well as different

[1] The utility whose network we monitored is small and part of a Utility Cooperative, and the control actions are not representative of larger, more modern, utilities.

software (operating systems, protocol stack implementations, number of measurements being taken, complexity of control logic) all resulting in each one being able to process requests at different speeds. However most importantly, no matter what kind of ICS network it is in or what physical value the device is measuring (e.g. voltage, pressure, flow rate, temperature) the device is still going to go through the same process of parsing the data request, retrieving the measurement from memory, and sending the response. Therefore, due to the limited processing power and fixed CPU load CLRTs can be leveraged to identify ICS device types, but this does not explain why the CLRTs are so constant over the network.

Network Level Characteristics. Although the use case for this technique (as in the deployment of any anomaly based IDS) would involve a training period on each target network, one of the desired properties for device fingerprinting in general is that the network architecture of the target not be a significant factor.

In a traditional corporate network mobile phones and laptops are constantly moving around and connecting to different wireless access points. The traffic they are generating is traveling over vast distances, encountering routers that are experiencing unpredictable loads, and consecutive packets are never guaranteed to take the same path over the Internet. However, devices in ICS networks are dedicated to one critical task and are fixed in a permanent location. The traffic generated from their regular polling intervals travel over relatively short geographic distance and over simple network architectures that offer little to no chance for consecutive packets to take different paths. The regular polling cycle means that routers and switches on ICS networks have consistent predictable loads which result in consistent and predictable queuing delays. Consequently for any given ICS network, a TCP ACK and SCADA response sent in quick succession will with extremely high probability take the same exact path, encounter the same delay, and therefore have a very consistent spacing in between them. Therefore, there is little opportunity for differences in network architecture to cause significant changes in the distribution of CLRTs. In Section IV-A3 we study how much a change in networks effects the performance by learning fingerprints from one substation and testing the fingerprints over a year later on a different substation.

Due to the low computational power found in ICS devices, the CLRTs are much larger than most delays that might be caused by differences in network architecture. In the real-world dataset used for this research, illustrated in Figure 6a, the CLRTs are all on the order of tens or even hundreds of milliseconds. In contrast, typical latencies obtained from ICS network switch datasheets and theoretical transmission delays on a 100Mbps link are both on the order of microseconds, resulting in a minor contribution to the overall CLRT measurement. Furthermore, ICS networks most often have over-provisioned available bandwidth to ensure reliability (e.g. the live power substation network studied for this research used an average of 11Kbps bandwidth out of the available 100Mbps, a strikingly low traffic intensity of 0.01%). These low traffic intensities ensure that the switches and routers on the network are never heavily loaded and have consistently low queuing delays.

Finally, even in the scenario where two network architectures are so different as to significantly alter the distribution of CLRTs, this would have no significant effect on the defensive utility of the proposed method and would arguably make it stronger. Any real-world application of the fingerprinting technique would involve a training period on the target network that would capture the minor effects of the network architecture. Then, if an attacker was attempting to create an offline database of signatures for all device types and software configurations without access to the specific target network, she would also have to consider all the possible network architectures that could affect them.

Due to this combination of low computational power, fixed CPU loads, and simple networks with predictable traffic, any significant change in a device's distribution of CLRTs highly suggests either an attacker spoofing the responses with a different machine, or a change in CPU workload [25] or software configuration, which could be a sign of a device being compromised with malware.

2) Experimental Setup: To test this method, experiments were run at a large scale using a real world dataset as well as on a small scale using controlled lab tests. The large scale tests were conducted in two rounds, before and after changes in the network architecture. First, network traffic (20GB) was captured from a live power substation with roughly 130 devices running the DNP3 protocol over the span of five months with the network architecture as illustrated in Figure 4. Then over a year later, one more month of data was captured from the same substation after the network was slightly modified by replacing the main router with a new switch, changing the IP addressing scheme accordingly, and increasing the frequency of measurement polling. Additionally, a brief overnight capture was collected from another substation with a different architecture (roughly 80 devices using DNP3, illustrated in Figure 5) to test if fingerprints learned on one network would translate to another. The company operating the substations provided a list of all device IP addresses on the network organized by location, device type, and device software configuration, and machine learning techniques were applied to attempt to make these labeled classifications.

Further tests were conducted in the lab to study the effects of the software configuration alone and to rule out any possible factors related to different hardware or different round-trip times (RTT) on the network.

In both scenarios, CLRT measurements were taken from DNP3 polling requests for event data and were summarized by dividing all measurements into time slices (e.g. one hour, or one day) and calculating means, variances, and 200-bin histograms for each time slice. Machine learning techniques were then evaluated using two different feature vectors: a more complex approach using the arrays of bin counts as defined in Equation 1 and a simple approach using arrays containing only the mean and variance for each time slice.

3) Results: **Device and Software Type Fingerprinting.** To obtain a rough visualization of the separability of the device types based on their CLRT measurements, a scatter plot based on the mean and variances of CLRTs was produced and the true labels of the devices were illustrated in Figure 6a. Each point represents the mean and variance of the CLRT

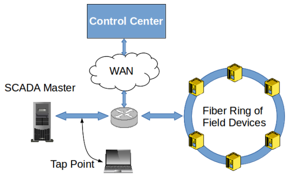


Fig. 4. Network Architecture of First Substation

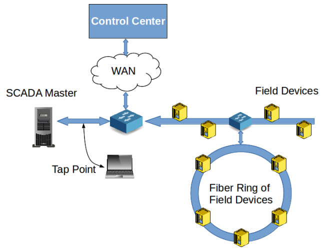


Fig. 5. Network Architecture of Second Substation

measurements for one IP address over the course of one day out of the original five month dataset. From the figure we can tell that even using simple metrics such as means and variances, results in the vendors and hardware device types being highly separable. Furthermore, it suggests that this method can also subdivide identical hardware device types into classes based on different software configurations (Vendor A Types 1a and 1b). For verification of this hypothesis, see Appendix A. These conclusions were further supported when the probability density functions (PDFs) of CLRTs over a day were estimated for each type in Figure 6b.

Since Figure 6a illustrates that device types are clearly separable based on simple mean and variance measurements, virtually any choice of a properly tuned machine learning algorithm would result in high accuracy classification. Therefore, as the purpose and novelty of this work is not the use of machine learning for fingerprinting, a sampling of the most popular algorithms in the field were chosen as examples.

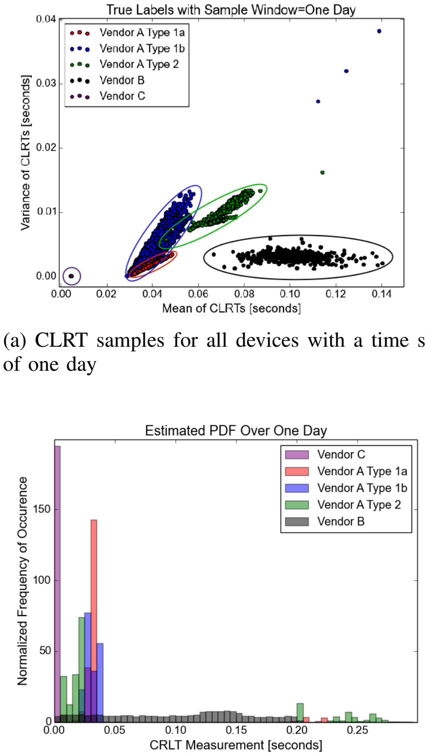
To measure the performance of our fingerprinting techniques throughout this work, we calculate the standard classification metrics of accuracy, precision, and recall as defined in Equations 2, 3, and 4 for each class separately, where TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. To summarize these results, the average value across classes was plotted alongside the minimum value among classes.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$PREC = \frac{TP}{TP + FP} \tag{3}$$

$$REC = \frac{TP}{TP + FN} \tag{4}$$

The first machine learning algorithm used in these experiments to classify the labeled data was a feed forward artificial neural network (FF-ANN) with one hidden layer trained using the back propagation algorithm. This algorithm was chosen due to its popularity and previous use in related work [18]. The bin counts of the histograms, as defined in Equation 1, were used as the feature vector for each sample and the time slice they were taken over was varied. The samples were randomly divided using 75% as training data and 25% as testing data. The average and minimum accuracy, precision, and recall for these experiments are shown in Figure 7, and suggests that even with time slices as small as 5 minutes an average accuracy of 93% can be achieved. Some devices at this substation were being polled only once every 2 minutes, so the 5 minute detection time is roughly equivalent to a decision after only two samples. Furthermore, when false data is injected into a control system catastrophic damage usually cannot immediately occur due to built-in safety features in the system. The most successful attacks would sabotage equipment or product over an extended period of time, for example by tricking a control system into heating a reactor past its limits and causing it to explode.



(a) CLRT samples for all devices with a time slice of one day

(b) Estimated PDFs of CLRTs for five sample devices over one day

Fig. 6. Separability of device types based on CLRT

To demonstrate that the exact choice of machine learning algorithm is largely irrelevant, we also attempted supervised learning using one of the simplest algorithms in the literature, a multinomial naïve Bayes classifier. The signature vectors remained the same and similar experiments were conducted to determine the required training period and detection time. Furthermore, these tests were conducted to simulate a real world deployment instead of randomly choosing training and test data, the training data was taken from the beginning of the capture and the test data was taken from the following 1000 detection time windows. After studying Figures 8a and 8b, it is clear that the simple Bayes classifier performs even better

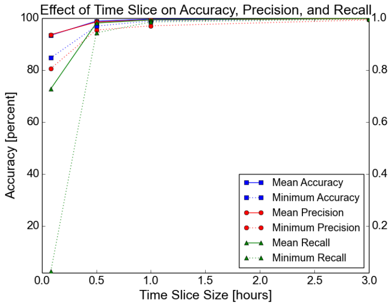
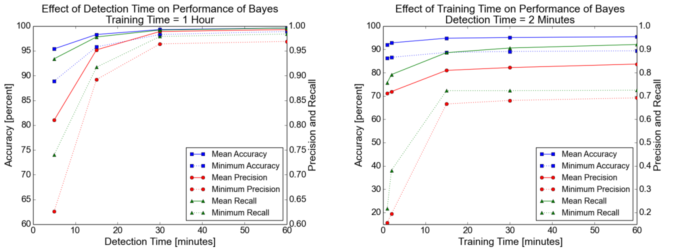


Fig. 7. Fingerprint Classification Performance Using FF-ANN



(a) Accuracy, precision, and recall of supervised Bayes classifier as a function of detection time

(b) Accuracy, precision, and recall as a function of training time

Fig. 8. Fingerprint classification performance

than the more complex ANN and can achieve high accuracy classification with detection times as small as a few minutes.

The above results are extremely promising for supervised learning when a list of IP addresses and corresponding device types are available, but this is not the case for administrators trying to understand what devices are on a poorly documented legacy network. To address this scenario, unsupervised learning techniques were also applied and tested if they could accurately cluster the devices into their true classes. Referring back to Figure 6a, it is clear that the samples closely follow a multivariate Gaussian distribution, so it was decided to illustrate unsupervised learning with Gaussian mixture models (GMM) using a full covariance matrix and a signature vector consisting of means and variances with a time slice of one day. Figure 9 shows the estimated clusters learned from the GMM algorithm, which upon comparison with the true clusters in Figure 6a, looks very similar. When the dataset was tested against the learned clusters, the model achieved an accuracy of 92.86%, a precision of 0.891, and a recall of 0.956. With performance as nearly as high as the supervised learning methods, this unsupervised technique would allow administrators to develop an accurate database of fingerprints with very little knowledge of the network itself.

Effect of Network Architecture. While the previous experiments, simulating a real-world deployment with a training period on the target network, performed very well, we also wanted to study how much the network architecture affects the performance of the fingerprinting techniques. For the first experiment to study these effects, the original substation was revisited over a year later after the network architecture had been upgraded and polling frequency had been increased. When the distribution of the new architecture in Figure 10a is compared with the original in Figure 6b, there are only minor differences. When the fingerprints learned from the original capture were tested on the new data, very high accuracies in Figure 11a were obtained suggesting that the method is stable over long periods of time and over minor changes in the same network.

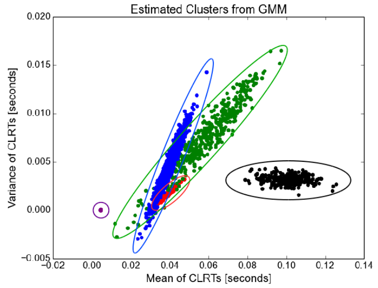
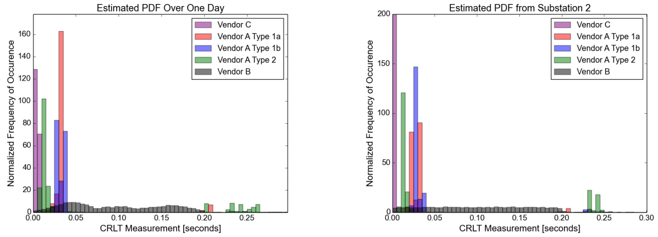


Fig. 9. Randomly generated samples from the unsupervised learned clusters



(a) CLRT Distribution After network Changes

(b) CLRT Distribution of Second Substation

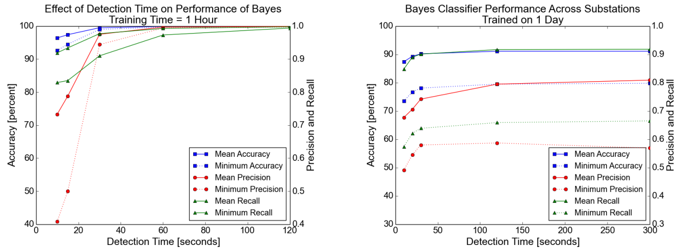
Fig. 10. Minor effects of network architecture on CLRT distributions

Even though the primary defensive use-case for this technique would always involve a training period on the target network, we also consider the rare case where an administrator is able to learn fingerprints on one network because of known labels, but does not have the labels for a different network. To study this scenario, we learned fingerprints from our original capture and tested them on a different substation over a year later. When the different substation's distribution in Figure 10b is compared with the original there are some small, but noticeable changes that could be result of the different architecture affecting the timings or from the different electrical circuit affecting the load of the devices. When the fingerprints learned from the original capture were tested on this different network, the average accuracy seemed to level off around 90% suggesting that while the accuracy may be diminished across different networks, there is still some utility in the technique.

Finally, to show that the technique performs well on different networks when trained individually, we trained a Bayes classifier on one hour of data from the second substation and tested it on the remaining seventeen hours of data with the results in Figure 12.

B. Method 2: Physical Fingerprinting

The second proposed fingerprinting approach addresses the control half of SCADA systems by fingerprinting physical devices based on their unique physical properties. A series of operation time measurements are taken and used to build an



(a) Classification Performance After Network Changes

(b) Classification Performance Across Substations

Fig. 11. Classification Performance Across Networks

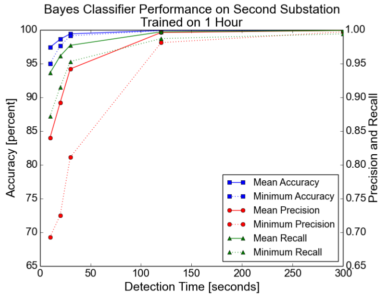


Fig. 12. Classification Performance on Second Substation

estimated distribution and generate the signature in a similar way as the first method. The formal definition of the signature in this case follows the same logic as Equation 1 above, but with M being defined as a set of operation time measurements and H being a heuristic threshold chosen to be an estimate of the maximum value an operation should ever take.

1) Theory: The mechanical and physical properties defining how quickly a device operates differs between devices and produces a unique fingerprint. For example, this concept is demonstrated by analyzing the difference in operation times of a latching relay that uses a solenoid coil arrangement illustrated in Figure 13. Relays were chosen for this research as they are commonly used in ICS networks for controlling and switching higher power circuits with low power control signals. The electromagnetic force produced while energizing the solenoid coil in a latching relay is directly proportional to current though the solenoid, number of turns in the solenoid, and the cross sectional area and type of core, as described by Equation 5.

$$F = \frac{(N * I)^2 u_0 A}{2g^2} \tag{5}$$

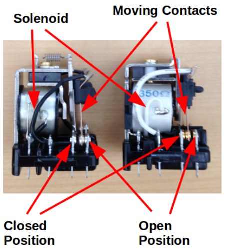


Fig. 13. Diagram of two different latching relays

N - Number of turns in the solenoid
I - the current, in amperes (A),running through the solenoid
A - the cross-sectional area, in meters-squared, of the solenoidal magnet
g - the distance in meters, between the magnet and piece of metal
μ_0 - $4\pi * 10^{-7}$ (a constant)

This electromagnetic force governs the operation time, and modification of any one of these variables due to differing vendor implementations results in unique signatures. In addition to proposing a specific distribution for devices based on vendor, individual physical operations like open or close will also produce a difference in operation times, which again can be attributed to the different forces involved in completing the physical action.

When a breaker or relay responds to an operate command from the master, an event change is observed at the slave device. With unsolicited responses enabled in the slave device, it asynchronously responds back with a message on an event change, which can be observed with a network tap to calculate the operation time. The response can also contain a sequence of event recorder (SER) timestamp indicating the time that the event occurred. Therefore, operation times can be estimated based on two different methods:

1) Unsolicited Response Timestamps - Calculated by the OS at the tap point by taking the difference between the time at which the command was observed and the time at which the response was observed. $m = t_3 - t_1$
2) SER Response Timestamps - Calculated from the difference between the time at which the command was observed at the tap point and the application layer event timestamp. $m = t_2 - t_1$

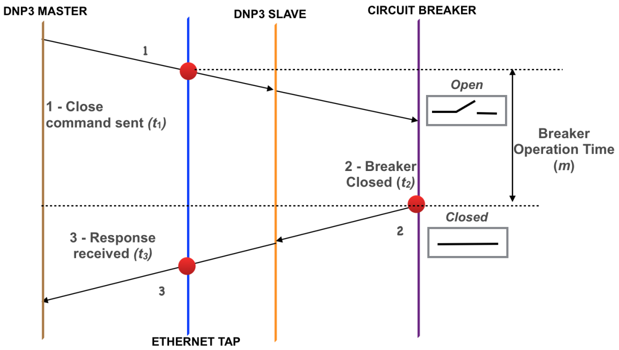

DNP3 MASTER
DNP3 SLAVE
CIRCUIT BREAKER
1
Open
1 - Close command sent (t_1)
Breaker Operation Time (m)
2 - Breaker Closed (t_2)
Closed
2
3 - Response received (t_3)
3
ETHERNET TAP

Fig. 14. Timing diagram to calculate Operation times

2) Experimental Setup: To demonstrate the proposed approach, the circuit breaker operation was chosen. The experimental setup consisted of a DNP3 master from a C++ open source DNP3 implementation (OpenDNP3 version 2.0), an SEL-751A DNP3 slave and two latching relays to demonstrate fingerprinting based on operation time. At the tap point in Figure 15, a C based DNP3 sniffer is used to sniff and parse the DNP3 packets to perform deep packet inspection. At the same tap point, the packets are timestamped by the Linux operating system which is time synchronized by the same time source as that of the DNP3 master and DNP3 slave.

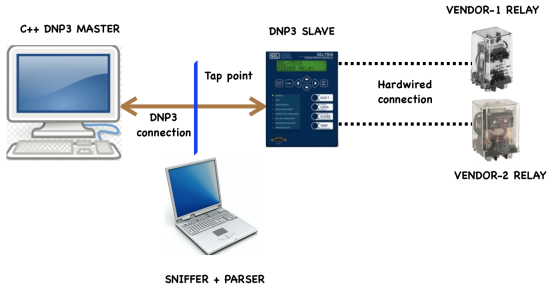
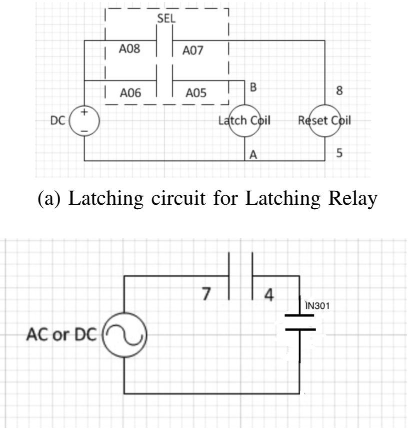


Fig. 15. Experimental Test Setup-fingerprinting breakers

The SEL-751A IED is a feeder protection relay supporting Modbus, DNP3, IEC61850 protocol, time synchronization based on SNTP protocol, and a fast SER protocol which timestamps events with millisecond resolution. The experimental setup for both relays consisted of a latching circuit (Figure 16a) and a load circuit (Figure 16b).



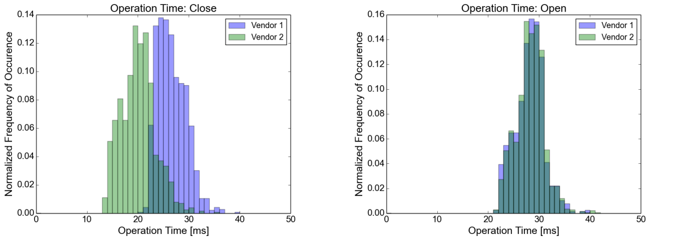
(a) Latching circuit for Latching Relay

(b) Load circuit for Latching Relay

Fig. 16. Circuits used in lab experiments

The latching circuit works on an operating voltage of 24VDC needing about 1A to operate and load circuit is based on 110V to be compatible with the IED's inputs. On a close command from the DNP3 master, the IED activates a binary output energizing the latch coil to close the load circuit. Once the load circuit is energized, the binary input senses the change and a timestamped event is generated. On an open command from the DNP3 master, the IED activates the second binary output energizing the reset coil to open the load circuit, which is recorded as a timestamped event. For these experiments, 2500 DNP3 open and close commands were issued simultaneously to both the latching relays with an idle time of 20 seconds between operations. The commands and responses were recorded at the tap point and operation times were calculated using both the unsolicited response method and SER based method. The unsolicited response method did not produce any usable results, so the SER method results are described below and retained as the physical fingerprint.

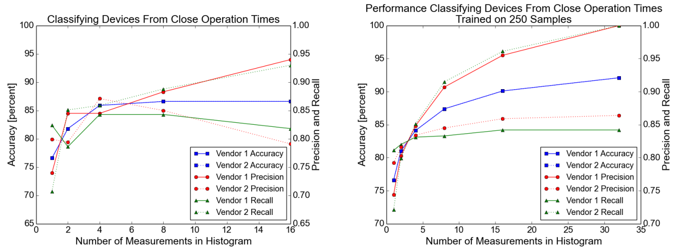
3) Results: **Difference between Vendors.** The distributions of close operation times based on SER timestamps for devices from two different vendors are illustrated in Figure 17a. The times range from 16ms to 38ms for Vendor 1 and 14ms to 33ms for Vendor 2. Even though both devices have similar ratings, the difference in operation can be attributed to the difference in physical makeup between them. For example, one device had a larger cross sectional area for its solenoid, resulting in different forces produced by Equation 5 above. When the same FF-ANN techniques as the first method were applied to classify the latches based on SER timestamped operations, the accuracy leveled off around 86% as shown in Figure 18a. Note that the large fluctuations appear to be a result of overfitting, causing one class's performance to improve significantly at the cost of the other.



(a) Distribution of Close Operation times based on SER Responses

(b) Distribution of Open Operation times based on SER Responses

Fig. 17. SER based response times



(a) FF-ANN Classifier Performance

(b) Bayes Classifier Performance

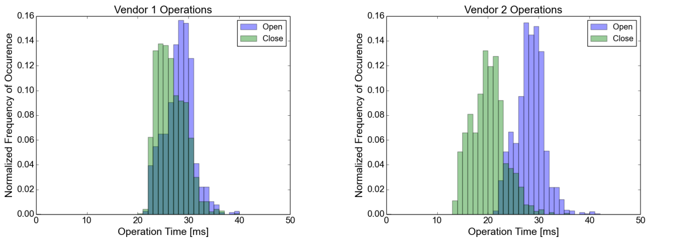
Fig. 18. Classification Performance Based on Timestamped Close Operations

When the naïve Bayes classifier was applied to this problem slightly better results were obtained in Figure 18b that leveled off around 92% accuracy, again suggesting that any properly tuned machine learning algorithm would perform well.

Figure 17b illustrates the distribution of open operation times for the two different latches and shows little variation between the two, thus preventing these times from being used for accurate device fingerprinting.

Difference between Operations. The previous results found that Close operation times help distinguish between relays of two different vendors, but it would also be desirable to distinguish between types of operations for a single device, for example to determine if a device had opened or closed in response to a command. Figure 19b shows the distribution of open and close operations for Vendor 1's latching relay with noticeable differences. These differences can be attributed to the physical construction of the components that act to open or close the relay, as discussed in detail in Section V.

On repeating the experiments for the second vendor's relay, the distribution of open and close operation times (Figure 19a)



(a) Distribution of open and close Operation times based on SER responses for Vendor 1

(b) Distribution of open and close Operation times based on SER responses for Vendor 2

Fig. 19. Difference between open and close

again showed clear distinctions and similar conclusions can be drawn as to the underlying causes. Therefore, even though the Open operation does not help distinguish between two vendors in this case, the results suggest that in the general case operations are distinguishable from one another and could potentially be used in other scenarios.

V. Synthetic Fingerprint Generation

While the results obtained in the previous section for both fingerprinting techniques (cross-layer fingerprinting and physical fingerprinting) are promising, the fingerprints were generated using black box methods that assume some access to the target devices. The first proposed technique based on monitoring of data packets requires a black box modeling approach as neither the internal circuitry nor the device source code is usually available (and thus there is no basis for constructing a white box model). Alternatively, physical fingerprinting technique may leverage a white box, black box, or gray box modeling approach since the mechanical composition of a device can usually be obtained from manual inspection, available drawings/pictures, or manufacturer's specifications. The ability to construct white box model fingerprints for physical fingerprinting is crucial due to the rare operation of some devices, and the prohibitive cost of performing black box modeling on all of the available devices on the market. To illustrate this technique, this section describes construction of the same fingerprint for the latch relay mechanism discussed in Section IV-B2 *using white box modeling only and then validates it against the black box model results obtained for the device in Section IV-B3*. However, a gray box modeling approach could be pursued as a general methodology for physical signature generation.

Modeling and Fingerprinting of a Latch Relay. To demonstrate the physical fingerprinting process, we consider a standard latch relay such as the Potter and Brumfield KUL Series relay shown in Figure 20 (Vendor 1 from the previous section). This latch relay operates using the principle of remanent magnetization in which a coil magnetizes a permanent magnet in either direction during opening and closing operations. To construct a dynamic model for the device, the latch relay was disassembled and its basic components modeled as shown in Figure 20. A magnetic armature of length L is connected to the base assembly by a torsional spring of spring constant k. The torsional spring is preloaded so that it applies a torque which pushes the armature to the open position by default. A permanent magnet lies at a distance l along the armature and is assumed to exert a magnetic force F_p at a single point along the armature. Furthermore, the permanent magnet is surrounded by a wire coil which carries the input current $\alpha(t)$, and also applies a magnetic force F_c to the armature. The magnetic field from the coil pulse drives the magnetic field of the permanent magnet to be in the same direction. After the driving field is removed, the permanent magnet holds the field in the same direction by the property of remanent magnetization. This process is what "latches" the relay.

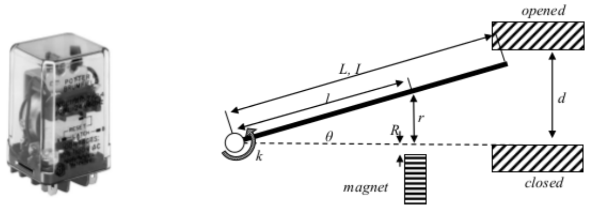


Fig. 20. Potter and Brumfield Latch Relay (left), Mechanical Schematic of Relay (right)

To switch the latch relay, a current is applied to the coil surrounding the permanent magnet. Let this current be given by the first-order response,

$$\alpha(t) = 1 - e^{-t/\tau} \tag{6}$$

where $t = 0$ corresponds to the time the switching command is initiated and τ is an appropriate time constant. The magnetic field produced by the coil induces a change in the magnetic field properties of the permanent magnetic through remanence [7]. To model this process, consider the function $\phi(t)$ given by,

$$\phi(t) = \frac{2}{\pi}\tan^{-1}(\beta\alpha(t) - \gamma) \tag{7}$$

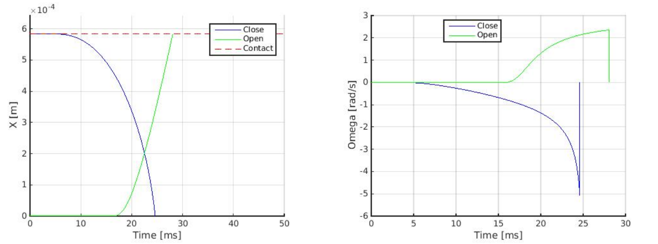
which approximately models the magnetic field of the permanent magnet as the current in the coil changes with time (where β and γ are tuning parameters). Given this approximation of the magnetic field, the forces exerted on the armature by the permanent magnet and coil are given respectively by,

$$F_p = \frac{c_p\mu_0}{(r+R)^2}\phi(t) \qquad F_c = \frac{c_c\mu_0}{(r+R)^2}\alpha(t) \tag{8}$$

where c_p and c_c are constants describing the strength of the magnet and μ_0 is the magnetic permeability of air. The equation of motion for the armature is thus,

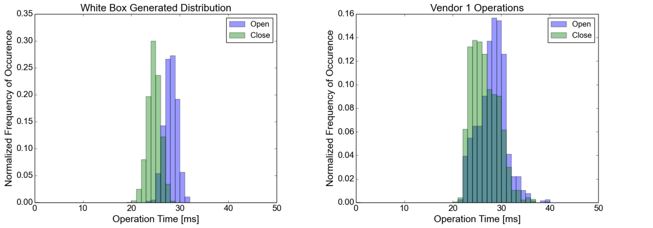
$$\ddot{\theta} = I^{-1}(F_p l\cos\theta + F_c l\cos\theta + k\theta) \tag{9}$$

where I is the moment of inertia of the armature about the hinge point. Physical measurements of the device can be used to provide values for r, R, l, L, k, and I. Five other parameters must be identified to simulate the time response of the latch relay mechanism, namely c_p, c_c, β, γ, and τ. These parameters may be estimated based on material composition of the magnets.



(a) Armature displacement vs time (b) Armature angular velocity vs time

Fig. 21. Armature displacement and angular velocity



(a) Simulated response times (b) Experimental response times (from Section IV-B3)

Fig. 22. Comparison of simulated and experimental distributions for the Potter and Brumfield KUL series latch relay

Figure 21 shows armature displacement and angular velocity time histories for an example opening and closing sequence, where displacement is measured at the contacts. Experimental data showed that the average opening time is longer than the average closing time which is reflected in simulation model outputs. Note that the simulation predicts that the opening and closing operations will take approximately 28 ms and 24 ms respectively under nominal conditions.

To generate a physical fingerprint (PDF), a Monte Carlo simulation was performed randomly perturbing the nominal values of the τ parameter using a Gaussian distribution. This data was compared with experimental results obtained using the setup described in Section IV-B2. Figure 22 shows a histogram of the response times for approximately 1200 runs, with simulated and experimental data shown on the left and right respectively. The similarity in these distributions demonstrates that the mechanical response characteristics can be adequately captured with this parameterized dynamic model. We extend this notion of white box modeling to a much larger and realistic power system device in Appendix B.

To test how well this white box modeled "synthetic signature" could be used in fingerprinting, the same machine learning techniques were applied as before, but trained from the simulated distribution for one device and experimental measurements from the other device. The FF-ANN was trained using the same number of samples for each device, and then performance was tested using an equal number experimental measurements for each device. With classification accuracy leveling off around 80% as shown in Figure 23, the white box model expectedly does not perform quite as well as the black box method based on true measurements due to the various simplifications and estimations made during the modeling process. However, the results are still very promising for this new class of fingerprinting. Furthermore, in a real world scenario the white box model approach would be limited to scenarios where there is not enough experimental data or the integrity of the experimental data is in question. The white box approach can then be combined with the black box approach to enable gray box modeling where appropriate to achieve higher accuracy. While there are a variety of techniques to approach this problem, Bayesian learning being one, intuitively it is similar to simply replacing synthetic samples in the white box distribution with real samples over time as they become available. Additional discussion of the limitations of white box modeling is provided in Section VI-C1.

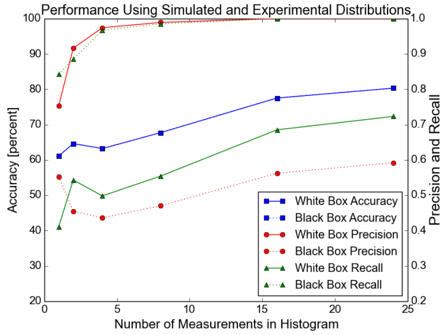


Fig. 23. Performance using a combination of white box and black box modeling

VI. Discussion

A. Performance

In order for a fingerprinting method to be useful for any situation, whether it is for intrusion detection, surveillance, or network management, the techniques should be relatively accurate and scalable.

Accuracy While neither method was able to obtain the near-perfect classification accuracy needed for an effective stand-alone intrusion detection system, both achieved high enough accuracy to prove useful in a defense-in-breadth strategy as a supplement to traditional IDS approaches. The CLRT method achieved impressive classification accuracies as high as 99% in some cases and the physical fingerprinting method was able to accurately classify measurements from two nearly identical devices around 92% of the time. For reference, all of the previous passive fingerprinting methods described in Section II achieved classification accuracies ranging from 86% to 100%, so these performances are quite comparable.

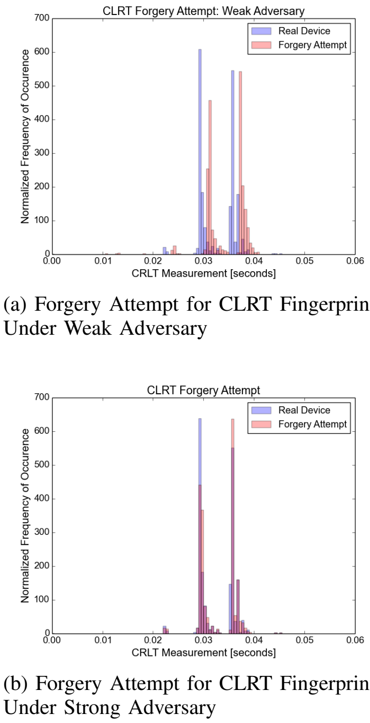
Scalability The FF-ANN algorithm used in training the two fingerprinting techniques only had one hidden layer and 200 input features, resulting in reasonable scalability for computational complexity, and the alternate Bayes classifier algorithm is also very efficient. Furthermore, our results suggest that the accuracy for the methods scales as well. The CLRT method was already tested above on a full scale power substation network and was able to achieve high accuracies. Although the physical fingerprinting method only achieved an accuracy of 92% for two similarly rated devices, it would be expected to achieve even higher accuracy as more diverse types of devices are added to the test set, resulting in more clear differences in distributions.

B. Robustness Against Forgery

When using device fingerprinting to augment traditional IDS methods, it is also desired that the fingerprints be nontrivial to forge (i.e., resistant to mimicry attacks). Fortunately there are several reasons as to why the proposed methods are not so easily broken. First, there is always going to be inherent randomness in the attacker's machine that makes it non-trivial to perfectly reproduce anything based on precision timing. Second, for the physical fingerprinting method the adversary machine's clock must stay synchronized with the target device's clock to millisecond precision. While this may not be very difficult with modern computers and networks, most devices in legacy control system networks have much lower powered processors and experience significant clock drift. For example in the observed dataset, the RTU (SCADA master for the field devices) drifted away from our network sniffer's clock at a rate of 6ms per hour.

To evaluate the proposed methods against forgery, we consider two different classes of adversary. First, we consider the case where an adversary is unable to gain physical access to the target network but instead is able to compromise one of the low powered devices on an air-gapped network, as in the case of Stuxnet [15]. Her goal is to watch the network long enough to generate black box fingerprints and spoof the responses of another device while matching their fingerprint. To model this adversary, we use a BeagleBone Black with 512MB of RAM and its ARM processor clocked down to 300MHz to simulate the resources available on a high-end PLC. Second, we consider a stronger adversary that has gained physical access to the network and is able to use her own, more powerful, machine to spoof the responses. This stronger adversary was modeled by a standard desktop with a 3.4 GHz quad-core i7 processor and 16GB of RAM. In both scenarios, the adversary is assumed to have gathered accurate samples and therefore has perfect knowledge of the signature she must try to mimic. However, in reality there are several difficulties that would make this perfect knowledge unlikely. First, since the ICS environment contains an abundance of legacy devices, it is not certain that the compromised device would even have a network card that supports promiscuous mode for network sniffing. Additionally, any sniffing code installed on a low powered, compromised device would most likely be computationally expensive enough to skew timing measurements on the system. Furthermore, since it was found in Figure 11 that network architecture does have some effect on the fingerprint, this suggests that the adversary would have to sniff the network in the same location as the fingerprinter to get a completely accurate distribution, or be able to determine the effects of the network by other means.

1) Cross-Layer Response Time Forgery: To test the cross-layer fingerprinting method, an open source implementation of DNP3 (OpenDNP3 version 2.0.1) was modified to have microsecond precision sleep statements using the known CLRT distribution of one of the Vendor A Type 1b devices. The forgery attempt by the weaker adversary in Figure 24a shows very clear differences in the distributions due to the limited resources slowing the distribution down and adding its own randomness. The stronger adversary's forgery attempt can be seen in Figure 24b. Compared with the original, the two distributions are very similar but the forged one is slightly slower due to the adversary's own processing time.



(a) Forgery Attempt for CLRT Fingerprinting Under Weak Adversary

(b) Forgery Attempt for CLRT Fingerprinting Under Strong Adversary

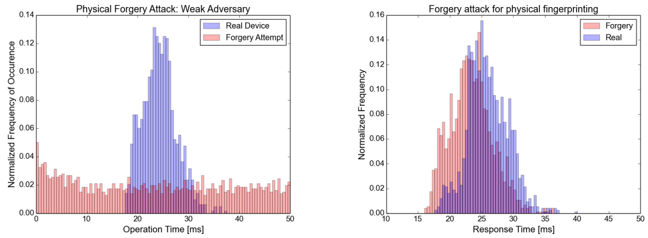
Fig. 24. Forgery attempts against the CLRT technique

When the Bayes classifier was applied to distinguish between the real device's distribution and the attacker's forged distribution, the results in Figure 26 suggest high accuracy detection of the forgery can be achieved.

2) Physical Fingerprinting Operation Time Forgery: To study the forgery of the physical fingerprinting technique, a DNP3 master was configured to send operate commands every second, and the adversary machine's modified OpenDNP3 code was programmed to send responses with timestamps calculated from the machine's current time, added with the known distribution of operation times. The resulting forgery attempt by the weaker adversary can be seen in Figure 25a. The distributions appear completely different due to the BeagleBone's clock quickly drifting from the SCADA master's, thus making the forgery attempt easily detected. The forgery attempt by the stronger adversary, illustrated in Figure 25b, is similar to the original, but still has noticeable differences most likely due to the high-end PC timestamping the operations faster than the original device.

The results from the Bayes classifier in this scenario in Figure 26 also suggest high accuracy detection of forgery is possible.

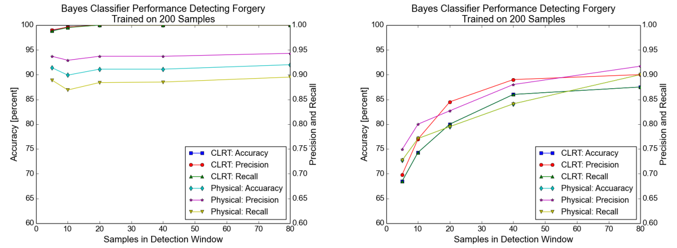
Even though both fingerprinting techniques exhibit resistance to these naïve forgery attacks, we admit it is still possible that an attacker could more intelligently shape her response times to more closely match the true fingerprint and implement a method of keeping better clock synchronization with the target. However, this would require a *significantly*



(a) Forgery Attempt for Physical Fingerprinting Under Weak Adversary

(b) Forgery Attempt for Physical Fingerprinting Under Strong Adversary

Fig. 25. Forgery attempts against the physical fingerprinting technique



(a) Forgery Detection for Weak Adversary

(b) Forgery Detection for Strong Adversary

Fig. 26. Forgery Detection

more knowledgeable and skilled adversary to successfully accomplish. She would have to know beforehand the relative speed of her machine to the target's machine, have knowledge of any effects the network architecture might have on the signature, and determine how fast the target's clock drifts, all suggesting that these methods are robust enough to be used as part of a defense-in-breadth IDS strategy.

Although the fingerprinting techniques proposed here are completely passive and require no changes to the target network or devices, better defenses against mimicry attacks could be implemented if this assumption is removed. For example, the SCADA master or the fingerprinter could be configured to randomly send extra requests or commands that have no effect on the operation of the network, but would increase the knowledge requirement of the adversary and the complexity of the behavior she has to mimic. For the CLRT method, this could involve changing from polling for event data to polling for different numbers of specific measurements each time, which on the low powered embedded systems should theoretically result in measurable timing differences. For the physical fingerprinting method this could take the form of sending redundant commands, for example by sending a Close command when the breaker is already closed.

C. Limitations

While both proposed fingerprinting methods perform well under certain conditions, there are some limitations. The cross-layer fingerprinting method first requires a SCADA protocol using "Read" and "Response" messages, which all of the most popular SCADA protocols implement. Furthermore, the SCADA protocol must sit on top of a TCP implementation that uses at least a minimum amount of "quick ACKs" (immediately ACKing a packet instead of delaying in the hopes of piggybacking). For example, modern Linux systems use quick ACKs to accelerate TCP slow start at the beginning of connections and after retransmissions, but every vendor in the observed power substation dataset used quick ACKs for every packet, presumably to reduce latency. Therefore, the amount of quick ACKs used by a device would determine how quickly a fingerprint could be generated.

The physical fingerprinting method requires high resolution timing of when operations take place, so it must be used with protocols that include operation timestamps in their responses. Not all SCADA protocol support this functionality, but the ones used in time-critical environments, such as the power grid, do include such timestamps. Requiring timestamps in the network traffic is a limitation in the sense that it can make it easier for an adversary to generate and forge the device fingerprints, but it can also be a defensive strength in another. If the network traffic is encrypted, an adversary would have to resort to white box modeling to attempt to generate any fingerprints, which is non-trivial and becomes more difficult as the devices modeled become more complex (e.g., Appendix B gives a coarse model of a more complex mechanical operation).

The highest classification accuracies achieved in this work, 99% and 92% for CRLT and physical fingerprinting respectively, are impressive but would result in an impractical number of false alarms (1% and 8%) if each mis-classification was treated directly as an intrusion. Therefore, any practical application of these fingerprinting techniques to detect intrusions would leverage the significant body of work [22] [8] on IDS alert correlation to manage the number of alarms.

1) Limitations of White Box Modeling: Clearly, the proposed white box modeling approach requires detailed knowledge of the mechanical construction of the ICS device. To construct a physics-based model, the devices basic mechanical functionality must be derived from either available schematics, drawings, or a physical example of the device itself. In some cases, the material composition of certain components (i.e., magnetic materials, etc) may also be important in the modeling process. For many devices this information is widely available and thus building a model is feasible. However, in some cases it is possible that mechanical design data will be difficult to obtain, for instance due to intellectual property concerns. For a given device, there is certain device-specific mechanical data that is required to build a physical model, and if this data is not available then white box modeling is likely infeasible.

Another consideration in white box modeling arises from process variation or model error. If the white box model exhibits parametric error only, Monte Carlo simulation can be used as in the above examples to generate a realistic response distribution by randomly varying model parameters. However, non-parametric modeling errors (or structural modeling errors) may pose significant problems as these can lead to bias errors in the resulting response distributions. Non-parametric errors may stem from unmodeled components or incorrect modeling assumptions. These biases in the model response can in turn lead to misclassification problems.

The most attractive method to mitigate structural model error is to employ a gray box modeling approach, which combines white box model predictions with black box data as it becomes available. For example, this can be accomplished by replacing synthetic samples with measured samples in the response distribution. The accuracy would then be expected to converge to black box model accuracy over time. It is important to note that, due to the accuracy limitations of white box modeling, use of this approach would be limited to scenarios where equipment is operated so infrequently that sufficient black box data is difficult to immediately obtain. White box modeling does, however, serve as a valuable tool in such scenarios by providing a starting guess for the response distribution that can be updated opportunistically as additional data is gathered.

VII. Conclusions and Future Work

In this paper we presented two novel methods for passively fingerprinting devices on ICS networks. After evaluating the methods using real world datasets and controlled lab experiments, fingerprint classification accuracies as high as 99% and 92% were achieved for the first and second methods respectively. Both techniques exhibited resistance to simple forgery attacks and could feasibly be implemented alongside more traditional IDS solutions to augment the security of critical ICS networks.

For future work, we plan to improve on the white box modeling and extend these methods to fingerprinting embedded devices in the "Internet of Things" and also investigate the possibility of developing active fingerprinting techniques to increase classification accuracy.

Acknowledgments

We thank the anonymous reviewers and our shepherd, Kevin Butler, for providing insightful feedback that helped improve the final paper.

References

[1] Advisory (icsa-15-041-02). https://ics-cert.us-cert.gov/advisories/ICSA-15-041-02.

[2] Nmap - free security scanner for network exploration & security audits. http://nmap.org/. Accessed 2015-03-25.

[3] M. Abrams and J. Weiss. Malicious control system cyber security attack case study-maroochy water services, australia. http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_report.pdf, 2008.

[4] A. Bates, R. Leonard, H. Pruse, D. Lowd, and K. Butler. Leveraging usb to establish host identity using commodity devices. In *Network and Distributed System Security (NDSS)*, NDSS '14, February 2014.

[5] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Fovino, and A. Trombetta. A multidimensional critical state analysis for detecting intrusions in scada systems. *Industrial Informatics, IEEE Transactions on*, 7(2):179–186, May 2011.

[6] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: Risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 355–366, New York, NY, USA, 2011. ACM.

[7] K. Davey. Calculation of magnetic remanence. *Magnetics, IEEE Transactions on*, 45(7):2907–2911, July 2009.

[8] H. Debar and A. Wespi. Aggregation and correlation of intrusion-detection alerts. In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, RAID '00, pages 85–103, London, UK, UK, 2001. Springer-Verlag.

[9] D. Formby, S. S. Jung, J. Copeland, and R. Beyah. An empirical study of tcp vulnerabilities in critical power system devices. In *Proceedings of the 2Nd Workshop on Smart Energy Grid Security*, SEGS '14, pages 39–44, New York, NY, USA, 2014. ACM.

[10] I. Fovino, A. Carcano, T. De Lacheze Murel, A. Trombetta, and M. Masera. Modbus/dnp3 state-based intrusion detection system. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 729–736, April 2010.

[11] J. Francois, H. Abdelnur, R. State, and O. Festor. Ptf: Passive temporal fingerprinting. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 289–296, May 2011.

[12] K. Gao, C. Corbett, and R. Beyah. A passive approach to wireless device fingerprinting. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 383–392, June 2010.

[13] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, 2(2):93–108, April 2005.

[14] O. Kosut, L. Jia, R. Thomas, and L. Tong. Limiting false data attacks on power system state estimation. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6, March 2010.

[15] R. Langner. To kill a centrifuge. http://www.langner.com/en/wp-content/uploads/2013/11/To-kill-a-centrifuge.pdf.

[16] H. Lin, A. Slagell, C. Di Martino, Z. Kalbarczyk, and R. K. Iyer. Adapting bro into scada: Building a specification-based intrusion detection system for the dnp3 protocol. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, CSIIRW '13, pages 5:1–5:4, New York, NY, USA, 2013. ACM.

[17] L. Ljung. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1 – 12, 2010.

[18] S. Radhakrishnan, A. Uluagac, and R. Beyah. Gtid: A technique for physical device and device type fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, PP(99):1–1, 2014.

[19] G. Shu and D. Lee. Network protocol system fingerprinting - a formal approach. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[20] S. Sridhar, A. Hahn, and M. Govindarasu. Cyber-physical system security for the electric power grid. *Proceedings of the IEEE*, 100(1):210–224, Jan 2012.

[21] O. Ureten and N. Serinken. Wireless security through rf fingerprinting. *Electrical and Computer Engineering, Canadian Journal of*, 32(1):27–33, Winter 2007.

[22] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer. Comprehensive approach to intrusion detection alert correlation. *Dependable and Secure Computing, IEEE Transactions on*, 1(3):146–169, July 2004.

[23] J. Verba and M. Milvich. Idaho national laboratory supervisory control and data acquisition intrusion detection system (scada ids). In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 469–473, May 2008.

[24] J.-W. Wang and L.-L. Rong. "cascade-based attack vulnerability on the us power grid ". *Safety Science*, 47(10):1332 – 1336, 2009.

[25] L. Watkins, W. Robinson, and R. Beyah. A passive solution to the cpu resource discovery problem in cluster grid networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(12):2000–2007, Dec 2011.

[26] M. Zalewski. p0f v3. http://lcamtuf.coredump.cx/p0f3/. Accessed 2015-03-25.

Appendix

A. Software Configuration Fingerprinting

To verify the suggestions from the large scale experiments that the software configuration was observable through CLRT measurements, lab experiments were performed on the same exact IED with different settings enabled and disabled. Approximately 700 CLRT measurements were taken for each of three cases: all extra settings enabled, only overcurrent protection enabled, and all extra settings disabled. When

TABLE I. VACUUM INTERRUPTER PARAMETERS

Parameter Name	Variable	Estimate
Viscous damping coefficient [kg/s]	c	*varied*
Contact separation [m]	d	0.010
Spring constant [N/m]	k	200.0
Contact to pivot distance [m]	l_{con}	0.0508
Spring to pivot distance [m]	l_{spr}	0.106
Spring offset [m]	l_{offset}	0.050
Total bar length [m]	l_{bar}	0.1568
Mass of contact [kg]	m_{con}	0.850
Mass of bar [kg]	m_{bar}	0.200
Normalized center of gravity [nd]	N_{CG}	0.500

comparing the distributions for all extra settings enabled versus disabled in Figure 27, there are several noticeable differences. In fact, when the same FF-ANN from the previous experiments was trained on these two cases, perfect classification accuracy was achieved. Figure 28 shows only minor differences between the 'free' case and the overcurrent case, and consequently, the FF-ANN only achieves roughly 66% classification accuracy.

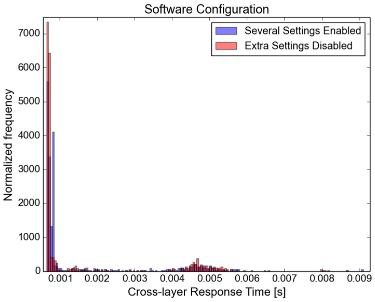


Fig. 27. Effect of multiple settings enabled on CLRT distribution

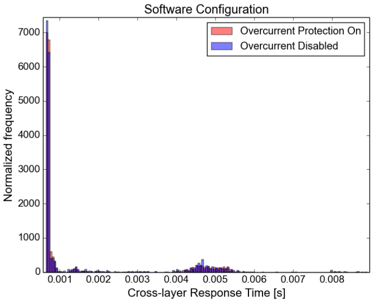
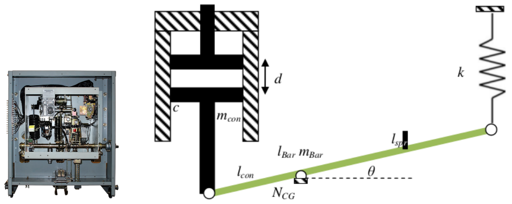


Fig. 28. Effect of one extra setting enabled on CLRT distribution

B. Modeling of a vacuum interrupter

The previous example in Section V highlights the modeling process for a small-scale relay for which laboratory data can be easily obtained. To demonstrate how this methodology scales to common ICS devices, the physical modeling approach is applied to a medium voltage vacuum circuit-breaker commonly found in power distribution stations. Vacuum interrupters typically employ contacts located inside a vacuum tube (used to mitigate arcing during operation). The breaker itself is a mechanical device operated by a preloaded spring so that opening and closing of the breaker happens rapidly. A picture of the Siemens GMSG vacuum circuit breaker and a mechanical schematic of the relevant moving parts are shown in Figure 29.



(a) Siemens GMSG medium voltage vacuum interrupter

(b) Schematic derived from the basic mechanical drawing

Fig. 29. Vacuum interrupter

With this schematic, the relevant equations of motion are found to be:

$$
\begin{aligned}
I\ddot{\theta} &= \mp(Flcos\theta)_{contact} - (Flcos\theta)_{spring} \\
I &= \frac{1}{12} m_{bar} l_{bar}^2 + m_{bar}(N_{CG} l_{bar} - l_{con})^2 + m_{con} l_{con}^2 \\
F_{contact} &= c l_{con} cos\theta\dot{\theta} \\
F_{spring,close} &= k\left[l_{spr} sin\theta + l_{offset}\right] \\
F_{spring,open} &= k\left[l_{spr} sin(\theta_{contact} - \theta) + l_{offset}\right]
\end{aligned}
\tag{10}
$$

After estimating the values of the parameters and varying the value of c in Monte Carlo simulations, the synthetic distribution in Figure 30 was generated. One interesting difference to note is that the open and close sequences are more similar to each other for the vacuum interrupter than they were for the latches. More importantly, it should be noted that this distribution is clearly distinguishable from the latches due to the significantly larger response times (centered around 60ms as opposed to 25ms).

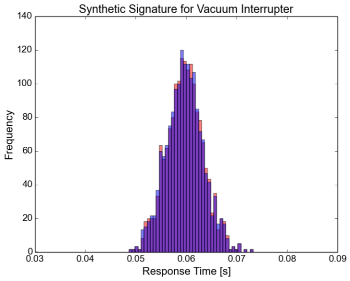


Fig. 30. Simulated open and close response distributions for vacuum interrupter

The purpose of the above example is not to provide a detailed mathematical analysis of a vacuum interrupter, but rather to demonstrate that a high-fidelity dynamic model of a real-world ICS component can be developed without requiring access to or operation of the device itself. Instead, available technical drawings and manufacturer's specifications are sufficient in many cases to estimate model parameters and generate a reasonable prediction of the device's response time distribution.